


```
PPPPPPPP      000000      000000      LL
PPPPPPPP      000000      000000
PP           PP   00       00   00       00   LL
PP           PP   00       00   00       00   LL
PP           PP   00       00   00       00   LL
PP           PP   00       00   00       00   LL
PPPPPPPP      00       00   00       00   LL
PPPPPPPP      00       00   00       00   LL
PP           00       00   00       00   LL
PP           00       00   00       00   LL
PP           00       00   00       00   LL
PP           00       00   00       00   LL
PP           00       00   00       00   LL
PP           00       00   00       00   LL
PP           000000      000000      LLLLLLLLLL
PP           000000      000000      LLLLLLLLLL
```

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
0001 0 %TITLE 'Dynamic memory management routines'
0002 0 MODULE POOL (IDENT = 'v04-000'
0003 0 %BLISS32[,ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE, NONEXTERNAL=LONG_RELATIVE)]
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1 FACILITY:
0032 1 DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility
0033 1
0034 1 ABSTRACT: Pooling for dynamic memory is supported.
0035 1
0036 1 ENVIRONMENT: Transportable
0037 1
0038 1 AUTHOR: R.W. Friday
0039 1
0040 1 CREATION DATE: January, 1979
0041 1
0042 1 MODIFIED BY:
0043 1 002 JPK00024 23-May-1983
0044 1 Modified lowercasing algorithm in PERMUTE. Now lowercase only
0045 1 if word contains only 1 letter or if second letter in word is
0046 1 lowercase. Picked up modules PAGMRG and POOL from DSR/DSRPLUS
0047 1 since they are no longer used by DSR/DSRPLUS.
0048 1
0049 1 --
0050 1
0051 1
0052 1 TABLE OF CONTENTS:
0053 1
0054 1
0055 1 FORWARD ROUTINE
0056 1 CPOOL : NOVALUE, ! Free all memory in a pool.
0057 1 FPOOL : NOVALUE, ! Free a pool and all storage.
```

POOL
V04-000

Dynamic memory management routines

J 8
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1 Page 2 (1)

```

58      0058 1      GPOOL : NOVALUE,      ! Get and initialize a pool
59      0059 1      IPOOL : NOVALUE,      ! Initialize an area as a pool control area.
60      0060 1      XPOOL;                  ! Extend a pool.
61      0061 1
62      0062 1
63      0063 1      INCLUDE FILES:
64      0064 1
65      0065 1
66      0066 1      LIBRARY 'NXPORT:XPORT';  ! XPORT macros
67      0067 1      REQUIRE 'REQ:DMDEFS';    ! POOL_...., PAD_...., & GET_SEG_ADDR def's
68      0148 1
69      0149 1
70      0150 1      MACROS:
71      0151 1
72      0152 1
73      0153 1
74      0154 1      EQUATED SYMBOLS:
75      0155 1
76      0156 1
77      0157 1      LITERAL
78      0158 1          TRUE = 1;
79      0159 1          FALSE = 0;
80      0160 1
81      0161 1
82      0162 1      OWN STORAGE:
83      0163 1
84      0164 1
85      0165 1
86      0166 1      EXTERNAL REFERENCES:
87      0167 1
```


POOL
V04-000

Dynamic memory management routines
CPOOL -- Free all memory in a pool

K 8
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1
Page 3
(2)

```

89 0168 1 %SBTTL 'CPOOL -- Free all memory in a pool'
90 0169 1 GLOBAL ROUTINE cpool (area) : NOVALUE =
91 0170 1
92 0171 1 ++
93 0172 1 FUNCTIONAL DESCRIPTION:
94 0173 1
95 0174 1 CPOOL deallocates all memory allocated to a pool.
96 0175 1
97 0176 1 FORMAL PARAMETERS:
98 0177 1
99 0178 1 AREA is the address of the pool.
100 0179 1
101 0180 1 IMPLICIT INPUTS:
102 0181 1
103 0182 1 NONE
104 0183 1
105 0184 1 IMPLICIT OUTPUTS:
106 0185 1
107 0186 1 NONE
108 0187 1
109 0188 1 ROUTINE VALUE:
110 0189 1 COMPLETION CODES:
111 0190 1
112 0191 1 NONE
113 0192 1
114 0193 1 SIDE EFFECTS:
115 0194 1
116 0195 1 NONE
117 0196 1
118 0197 1 --
119 0198 1
120 0199 2 BEGIN
121 0200 2
122 0201 2 BIND SEGMENT_LIST = ..AREA + POOL_CNTRL_SIZE*%UPVAL; !Point to start of segment list.
123 0202 2
124 0203 2
125 0204 2 BIND XAREA = ..AREA : POOL; !AREA is the address of the pool.
126 0205 2
127 0206 2
128 0207 2 !
129 0208 2 ! Don't try to clear an empty pool.
130 0209 2
131 0210 2 IF ..AREA EQL 0 THEN RETURN;
132 0211 2
133 0212 2 INCR I FROM 0 TO .XAREA [POOL_ACT_PADS] - 1 DO
134 0213 3 BEGIN
135 0214 3 BIND SEGMENT = SEGMENT_LIST + (.I*PAD_CNTRL_SIZE*%UPVAL) : PAD; !Point to a descriptor.
136 0215 3
137 0216 3
138 0217 3 !Free this chunk of memory.
139 P 0218 3 $XPO_FREE_MEM ( BINARY DATA =
140 P 0219 3 (.SEGMENT[PAD_SIZE], !Size of segment
141 0220 3 .SEGMENT[PAD_ADDRESS]) ); !Start of segment.
142 0221 3
143 0222 3 END;
144 0223 1 END; !End of CPOOL
```

POOL
V04-000

Dynamic memory management routines
CPOOL -- Free all memory in a pool

L 8
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1 Page 4
(2)

```

      001C 00000
54      04 5E      08 C2 00002
      BC      0C C1 00005
      53      04 BC D0 0000A
      52      34 13 0000E
      50      01 CE 00010
      60      2A 11 00013
      6E      64 42 7E 00015 1$:
      02 AE      04 A5 00019
      03 AE      02 90 0001D
      04 AE      02 90 00021
      04 AE      04 A0 D0 00025
      00000000G EF 9F 0002A
      7E      7E 7C 00030
      10      02 CE 00032
      00000000G EF 9F 00035
      D1      05 FB 00038
      52      04 A3 F2 0003F 2$:
      04      04 00044 3$:

```

.TITLE POOL Dynamic memory management routines
.IDENT \V04-000\

.EXTRN XPOS\$FREE_MEM, XPOS\$FAILURE

.PSECT \$CODE\$,NOWRT,2

.ENTRY CPOOL, Save R2,R3,R4

SUBL2 #8, SP

ADDL3 #12, @AREA, R4

MOVL @AREA, R3

BEQL 3\$

MNEGL #1, I

BRB 2\$

MOVAQ (R4)[I], R0

MULW3 #4, (R0), \$XPOS\$DESC

MOVB #2, \$XPOS\$DESC+2

MOVB #2, \$XPOS\$DESC+3

MOVL 4(R0), \$XPOS\$DESC+4

PUSHAB XPOS\$FAILURE

CLRQ -(SP)

MNEGL #2, -(SP)

PUSHAB \$XPOS\$DESC

CALLS #5, XPOS\$FREE_MEM

AOBLSS 4(R3), I, 1\$

RET

: 0169

: 0202

: 0205

: 0210

: 0212

: 0215

: 0220

: 0212

: 0223

; Routine Size: 69 bytes, Routine Base: \$CODE\$ + 0000

POOL
V04-000

Dynamic memory management routines
FPOOL -- Free a pool and all storage

M 8
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1
Page 5
(3)

```

: 146 0224 1 %SBTTL 'FPOOL -- Free a pool and all storage'
: 147 0225 1 GLOBAL ROUTINE fpool (area) : NOVALUE =
: 148 0226 1
: 149 0227 1 ++
: 150 0228 1 FUNCTIONAL DESCRIPTION:
: 151 0229 1
: 152 0230 1 Free an entire storage pool. All the areas pointed to by the active
: 153 0231 1 PADS are freed. Then the pool area itself is freed.
: 154 0232 1
: 155 0233 1 FORMAL PARAMETERS:
: 156 0234 1
: 157 0235 1 AREA is the address of the POOL
: 158 0236 1
: 159 0237 1 IMPLICIT INPUTS:
: 160 0238 1
: 161 0239 1 NONE
: 162 0240 1
: 163 0241 1 IMPLICIT OUTPUTS:
: 164 0242 1
: 165 0243 1 NONE
: 166 0244 1
: 167 0245 1 ROUTINE VALUE:
: 168 0246 1 COMPLETION CODES:
: 169 0247 1
: 170 0248 1 NONE
: 171 0249 1
: 172 0250 1 SIDE EFFECTS:
: 173 0251 1
: 174 0252 1 NONE
: 175 0253 1
: 176 0254 1 --
: 177 0255 1
: 178 0256 2 BEGIN
: 179 0257 2
: 180 0258 2 BIND !AREA is the address of the pool.
: 181 0259 2 XAREA = ..AREA : POOL;
: 182 0260 2
: 183 0261 2 CPOOL (.AREA); !First, deallocate all associated storate.
: 184 0262 2
: 185 0263 2 !
: 186 0264 2 ! Don't try to free the pool if it's empty.
: 187 0265 2 !
: 188 0266 2 IF ..AREA NEQ 0
: 189 0267 2 THEN
: 190 0268 2 $XPO_FREE_MEM ( BINARY_DATA = (.XAREA [POOL_ACT_SIZE], ..AREA));
: 191 0269 2
: 192 0270 2 .AREA = 0; !Clear pool pointer holder.
: 193 0271 1 END; !End of FPOOL
```

```

5E      0004 00000      .ENTRY FPOOL, Save R2      : 0225
52      08 C2 00002      SUBL2 #8, SP              :
      04 BC D0 00005      MOVL @AREA, R2             : 0259
      04 AC DD 00009      PUSHL AREA                  : 0261
```

POOL
V04-000

Dynamic memory management routines
FPOOL -- Free a pool and all storage

N 8
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1

Page 6
(3)

	AB	AF		04	01	FB	0000C	CALLS	#1, CPOOL		
					BC	D5	00010	TSTL	@AREA	:	0266
					27	13	00013	BEQL	1\$	:	
6E	08	A2			04	A5	00015	MULW3	#4, 8(R2), \$XPOSDESC	:	0268
	02	AE			02	90	0001A	MOVB	#2, \$XPOSDESC+2	:	
	03	AE			02	90	0001E	MOVB	#2, \$XPOSDESC+3	:	
	04	AE		04	BC	D0	00022	MOVL	@AREA, \$XPOSDESC+4	:	
			00000000G		EF	9F	00027	PUSHAB	XPOSFAILURE	:	
					7E	7C	0002D	CLRQ	-(SP)	:	
		7E			02	CE	0002F	MNEGL	#2, -(SP)	:	
				10	AE	9F	00032	PUSHAB	\$XPOSDESC	:	
00000000G	EF				05	FB	00035	CALLS	#5, XPOSFREE_MEM	:	
				04	BC	D4	0003C	CLRL	@AREA	:	0270
					04	00	003F	RET		:	0271

; Routine Size: 64 bytes, Routine Base: \$CODE\$ + 0045


```
195 0272 1 %SBTTL 'GPOOL -- Get and initialize a pool'
196 0273 1 GLOBAL ROUTINE gpool (area, count) : NOVALUE =
197 0274 1
198 0275 1 ++
199 0276 1 FUNCTIONAL DESCRIPTION:
200 0277 1
201 0278 1 GPOOL generates a pool area and initializes it.
202 0279 1
203 0280 1 FORMAL PARAMETERS:
204 0281 1
205 0282 1 AREA is set to the address of the generated pool.
206 0283 1 COUNT is the number of extensions to be supported.
207 0284 1
208 0285 1 IMPLICIT INPUTS:
209 0286 1
210 0287 1 NONE
211 0288 1
212 0289 1 IMPLICIT OUTPUTS:
213 0290 1
214 0291 1 NONE
215 0292 1
216 0293 1 ROUTINE VALUE:
217 0294 1 COMPLETION CODES:
218 0295 1
219 0296 1 Returns TRUE if generation was successful, otherwise FALSE.
220 0297 1
221 0298 1 SIDE EFFECTS:
222 0299 1
223 0300 1 NONE
224 0301 1
225 0302 1 --
226 0303 1
227 0304 2 BEGIN
228 0305 2 LOCAL
229 0306 2 SIZE;
230 0307 2
231 0308 2 SIZE = .COUNT*PAD_CNTRL_SIZE + POOL_CNTRL_SIZE; !Physical size of pool control area.
232 0309 2
233 0310 2 $XPO GET MEM (FULLWORDS =.SIZE, RESULT = .AREA); !Allocate pool control area.
234 0311 2 IPOOL(.AREA, .SIZE); !Initialize pool control area.
235 0312 1 END; !End of GPOOL
```

.EXTRN XPOSALLOC_MEM

```
0004 00000
5E 04 C2 00002
52 08 AC D0 00005
52 02 C4 00009
52 03 C0 0000C
02020000 8F DD 0000F
04 AE D4 00015
00000000G EF 9F 00018
7E 7E 7C 0001E
01 CE 00020
10 AE 9F 00023
```

```
.ENTRY GPOOL, Save R2
SUBL2 #4, SP
MOVL COUNT, R2
MULL2 #2, SIZE
ADDL2 #3, SIZE
PUSHL #33685504
CLRL $XPOS$DESC+4
PUSHAB XPOS$FAILURE
CLRQ -(SP)
MNEGL #1, -(SP)
PUSHAB $XPOS$DESC
```

: 0273

: 0308

: 0310

; Routine Size: 70 bytes, Routine Base: \$CODE\$ + 0085

POOL
V04-000

Dynamic memory management routines
IPOOL -- Initialize a pool control area

D 9
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1
Page 9
(5)

```

: 237 0313 1 %SBTTL 'IPOOL -- Initialize a pool control area'
: 238 0314 1 GLOBAL ROUTINE ipool (area, size) : NOVALUE =
: 239 0315 1
: 240 0316 1 ++
: 241 0317 1 FUNCTIONAL DESCRIPTION:
: 242 0318 1
: 243 0319 1     IPOOL initializes a pool area.
: 244 0320 1
: 245 0321 1 FORMAL PARAMETERS:
: 246 0322 1
: 247 0323 1     AREA is the address of the generated pool.
: 248 0324 1     SIZE is the number of BPVALS in the area.
: 249 0325 1
: 250 0326 1 IMPLICIT INPUTS:
: 251 0327 1
: 252 0328 1     NONE
: 253 0329 1
: 254 0330 1 IMPLICIT OUTPUTS:
: 255 0331 1
: 256 0332 1     NONE
: 257 0333 1
: 258 0334 1 ROUTINE VALUE:
: 259 0335 1 COMPLETION CODES:
: 260 0336 1
: 261 0337 1     NONE
: 262 0338 1
: 263 0339 1 SIDE EFFECTS:
: 264 0340 1
: 265 0341 1     NONE
: 266 0342 1
: 267 0343 1 --
: 268 0344 1
: 269 0345 2 BEGIN
: 270 0346 2
: 271 0347 2 BIND
: 272 0348 2     XAREA = ..AREA : POOL;
: 273 0349 2
: 274 0350 2
: 275 0351 2     XAREA [POOL_MAX_PADS] = (.SIZE - POOL_CNTRL_SIZE)/PAD_CNTRL_SIZE;
: 276 0352 2     XAREA [POOL_ACT_SIZE] = .SIZE;
: 277 0353 2     XAREA [POOL_ACT_PADS] = 0;
: 278 0354 1 END;

```

!Maximum number of areas.
!Physical size of pool control area.
!No areas yet.

!End of IPOOL

				0000	00000
			04	BC	D0 00002
51	08	50		03	C3 00006
60		AC		02	C7 0000B
	08	51		AC	D0 0000F
		AO	08	A0	D4 00014
			04		04 00017

```

.ENTRY IPOOL, Save nothing
MOVL @AREA, R0
SUBL3 #3, SIZE, R1
DIVL3 #2, R1, (R0)
MOVL SIZE, 8(R0)
CLRL 4(R0)
RET

```

: 0314
: 0348
: 0351
: 0352
: 0353
: 0354

; Routine Size: 24 bytes, Routine Base: \$CODE\$ + 00CB

POOL
V04-000

Dynamic memory management routines
IP00L -- Initialize a pool control area

E 9
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1 Page 10
(5)

PU
VO

POOL
V04-000

Dynamic memory management routines
XPOOL -- Extend a pool

F 9
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1 Page 11
(6)

```

280 0355 1 %SBTTL 'XPOOL -- Extend a pool'
281 0356 1 GLOBAL ROUTINE xpool (area,size) =
282 0357 1
283 0358 1 ++
284 0359 1 FUNCTIONAL DESCRIPTION:
285 0360 1
286 0361 1 XPOOL extends a dynamic storage pool by allocating a
287 0362 1 block of dynamic storage and chaining it into the pool.
288 0363 1
289 0364 1 FORMAL PARAMETERS:
290 0365 1
291 0366 1 AREA is the address of the pool to be extended.
292 0367 1 SIZE is the number of BPVALS needed.
293 0368 1
294 0369 1 IMPLICIT INPUTS:
295 0370 1
296 0371 1 NONE
297 0372 1
298 0373 1 IMPLICIT OUTPUTS:
299 0374 1
300 0375 1 NONE
301 0376 1
302 0377 1 ROUTINE VALUE:
303 0378 1 COMPLETION CODES:
304 0379 1
305 0380 1 Returns address of start of new segment if one could be
306 0381 1 could be allocated. Otherwise returns 0.
307 0382 1
308 0383 1 SIDE EFFECTS:
309 0384 1
310 0385 1 NONE
311 0386 1
312 0387 1 --
313 0388 1
314 0389 2 BEGIN
315 0390 2
316 0391 2 BIND
317 0392 2 XAREA = ..AREA : POOL; !AREA is the address of the pool.
318 0393 2
319 0394 2 IF .XAREA [POOL_ACT_PADS] EQL .XAREA [POOL_MAX_PADS] THEN RETURN 0;
320 0395 2
321 0396 2 BEGIN
322 0397 2 BIND
323 0398 2 NEW_PAD = ..AREA + %UPVAL*( (.XAREA [POOL_ACT_PADS] * PAD_CNTRL_SIZE) + POOL_CNTRL_SIZE ) : PAD;
324 0399 2
325 0400 2 !Allocate the segment.
326 0401 2 $XPO GET MEM ( FULLWORDS = .SIZE, RESULT = NEW_PAD [PAD_ADDRESS]);
327 0402 2 NEW_PAD [PAD_SIZE] = .SIZE; !Remember physical size of segment.
328 0403 2 !Update PAD count
329 0404 2 XAREA [POOL_ACT_PADS] = .XAREA [POOL_ACT_PADS] + 1;
330 0405 2
331 0406 2 !Tell user where the segment is.
332 0407 2 RETURN .NEW_PAD [ PAD_ADDRESS];
333 0408 2 END;
334 0409 2
335 0410 1 END; !End of XPOOL
```

				000C 00000	.ENTRY XPOOL, Save R2,R3		0356
5E			08	C2 00002	SUBL2 #8, SP		
52	04		BC	D0 00005	MOVL @AREA, R2		0392
50	04		A2	D0 00009	MOVL 4(R2), R0		0394
62			50	D1 0000D	CMP R0, (R2)		
			3D	13 00010	BEQL 2\$		
53	0C	A240	7E	00012	MOVAQ 12(R2)[R0], R3		0398
6E	02020000		8F	D0 00017	MOVL #33685504, \$XPOSDESC		0401
	04		AE	D4 0001E	CLRL \$XPOSDESC+4		
	00000000G		EF	9F 00021	PUSHAB XPOSFAILURE		
			7E	7C 00027	CLRL -(SP)		
7E			01	CE 00029	MNEGL #1, -(SP)		
	10		AE	9F 0002C	PUSHAB \$XPOSDESC		
7E	08		02	78 0002F	ASHL #2, SIZE, -(SP)		
	00000000G		06	FB 00034	CALLS #6, XPOSALLOC MEM		
			50	E9 0003B	BLBC \$XPOSSTATUS, T\$		
	04		AE	D0 0003E	MOVL \$XPOSDESC+4, 4(R3)		
		04	AC	D0 00043 1\$:	MOVL SIZE, (R3)		0402
		08	A2	D6 00047	INCL 4(R2)		0404
		04	A3	D0 0004A	MOVL 4(R3), R0		0407
			04	0004E	RET		
		50	D4	0004F 2\$:	CLRL R0		0410
			04	00051	RET		

; Routine Size: 82 bytes, Routine Base: \$CODE\$ + 00E3

:	336	0411	1	
:	337	0412	1	END
:	338	0413	0	ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	309	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_255\$DUA28:[SYSLIB]XPORT.L32;1	590	21	3	252	00:00.2

POOL
V04-000

Dynamic memory management routines
XPOOL -- Extend a pool

H 9
16-Sep-1984 01:28:47
14-Sep-1984 13:07:47

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]POOL.BLI;1
Page 13
(6)

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:POOL/OBJ=OBJ\$:POOL MSRC\$:POOL/UPDATE=(ENHS\$:POOL)
; Size: 309 code + 0 data bytes
; Run Time: 00:09.0
; Elapsed Time: 00:17.5
; Lines/CPU Min: 2747
; Lexemes/CPU-Min: 43124
; Memory Used: 63 pages
; Compilation Complete

0347 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY